卡弹问题——修改状态机防止烧电机

1. 问题

机器人发射通过状态机完成。但是状态机不够完善，当卡弹时，拨弹轮会持续供以大电流，烧电机。由于官方电机比较智能，当过热时会自动报警，因此目前还没有出现火灾。

1. 解决思路

在原先的状态机上增加状态转移。为发射一发的停留时间增加时间限制。时间到达后，转移到stuckSolve状态。stuckSolve状态下，电机持续以v/4持续反向转动一段时间。

1. 具体修改

在状态定义中增加stuckSolve状态。

//\*\*\*\*\*\*\*\*\*\*\*拨弹轮电机模式枚举类型

enum TriggerMode\_e{

    TriggerMode\_e\_Stop,         //停止发射

    TriggerMode\_e\_ShootOne,     //发射一个，靠拨弹轮减速前的电机转4圈。对应弹盘轮子转一格

    TriggerMode\_e\_ShootMulti,   //多发，其实是等待进入发射一个的状态

    TriggerMode\_e\_StuckSolve,   //尝试解决卡弹

};

增加时间定义

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*卡弹相关\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#define STUCK\_TIME\_LIMIT 1000   //当在发射一发的模式中持续一段时间不退出后，进入卡弹解决模式

#define SOLVING\_TIME\_LIMIT 500  //在尝试卡弹解决模式中停留的时间。

定义任务函数中的变量

    static uint32\_t shootOneEnterTime = 0;  //进入发射一发的状态的时刻

    static uint32\_t shootOneStayTime = 0;   // 在发射一发的状态中停留的时间

    static uint32\_t solveStuckEnterTime = 0;   // 在发射一发的状态中停留的时间

    static uint32\_t solveStuckNowTime = 0;   // 在尝试解决卡弹状态中停留的时间

修改状态机shoot one转移过程

else if(nowTime - shootOneEnterTime > STUCK\_TIME\_LIMIT)

        {

            triggerMode=TriggerMode\_e\_StuckSolve;   // 到达了卡弹的临界

        }

新增新的状态

else if(triggerMode==TriggerMode\_e\_StuckSolve)

    {

        if((nowTime - solveStuckEnterTime)>=SOLVING\_TIME\_LIMIT)

        {

            triggerMode=TriggerMode\_e\_Stop;

        }

    }

增加reverse定义（为了减小和之前的区别）

// 拨弹轮开启

uint8\_t triggerOn=0;

uint8\_t reverse = 0;

增加stuckSolve状态开启摩擦轮

if(triggerMode==TriggerMode\_e\_ShootOne || triggerMode == TriggerMode\_e\_StuckSolve)

        triggerOn=1;

    else

        triggerOn=0;

增加reverse状态

    reverse = 0;

    if(triggerMode == TriggerMode\_e\_StuckSolve)

        reverse = 1;

修改拨弹速度生成

if(triggerOn)

        wantedVTriggerMotor=reverse?(SHOOT\_TRIGGER\_SPEED\_LIMIT/4):(-SHOOT\_TRIGGER\_SPEED\_LIMIT);

    else

        wantedVTriggerMotor=0;

1. 实验结果

可以在卡弹后回弹。但是会连发有问题。是有个进入时间没有初始化。增加

if((nowTime-lastShootMultiTime)>=SHOOT\_MULTI\_TIME\_GAP)

        {

            lastShootMultiTime=nowTime;

            triggerMode=TriggerMode\_e\_ShootOne;

            shootOneEnterTime = nowTime;

        }

在所有进入shootOne 的地方都应该增加这个初始化进入时间的语句

后来发现卡弹恢复后，重新发射会有转不动的情况，是因为状态跳转时出现没有清空上一次的圈数

增加

else if(nowTime - shootOneEnterTime > STUCK\_TIME\_LIMIT)

        {

            triggerMode=TriggerMode\_e\_StuckSolve;   // 到达了卡弹的临界

            solveStuckEnterTime = nowTime;

            triggerCtrl.nowRounds = 0;//清空旋转量

        }

结果还不错，不会卡弹了。至少它会在卡的时候尝试回退并向前冲击，连发模式会持续直到发射出来。